Changes made in response to reviewer comments

Paper: SciSpot: Scientific Computing On Temporally Constrained Cloud Preemptible VMs

Our sincere thanks to all reviewers for their careful, insightful, and very helpful feedback and comments. Below, we summarize the major changes. In the rest of this document, we describe how we have addressed every comment (**in bold**), along with locations of the relevant changes in the paper text.
Summary of changes by section:

1. New introduction to highlight the contributions
2. Merged background and related work
3. New data analysis (2 graphs) added, subsection on model generalization,
4. Made workload and bags of jobs assumption clear. We do not need exact job running times for our policies.
5. Implementation role of bags of jobs, and hot-spares clarification.
6. Added details of experimental conditions, exosphere, and HPC waiting times.
7. Added discussion section
8. Added conclusion

Reviewer-1

1. The paper mentions that flat pricing is used in both GCP and Azure, but this paper has only considered GCP in the experiments. Can authors comment on the possibility of running the proposed framework on Azure or both together?
Revision made: Paragraph added in Discussion section. Azure does not have temporally constrained preemptions, so the modeling techniques would not be appropriate.

2. Does the bathtub-shaped preemption rate change day by day? In other words, is it necessary to measure the rate of preemption and update the parameters in the distribution function?
Revision made: We have added new temporal analysis of preemptions (Figure 3). In short, the preemption rate does change by day, but still remains bathtub shaped. We have also emphasized in Section 3 that we evaluate what happens when even a “rough” imperfectly fitted bathtub model is used (Figure 9). We have added a new subsection (3.4) addressing these concerns.

3. Based on the bathtub-shaped rate of preemption, perhaps a better scheme than that proposed in Sec. 4.2 is to maintain a pool of VMs “that have survived the initial failures”. VMs close to the end of the 24 hours can also be moved out of the pool directly. In this way, jobs do not need to choose VMs as all VMs are available. Does it make the scheduling/VM assignment easier than the method proposed in the paper?
Revision Made: Added details about this in implementation section. We indeed implement this technique and find it very helpful. We retain hot-spares for 1 hour.

There are also a few issues related to the presentation/writing of this paper:

1. It looks like the template is manually adjusted to shrink the space between the figure and the caption. Not sure if it is allowed…
Revision Made: Fixed Spacing

2. Conclusion of the paper is missing.
Revision Made: Added Conclusion

3. The symbol for the CDF function is inconsistent. \mathcal{F}(t) is used only in (1). In other parts of the paper, only F(t) is used.
Revision Made: Fixed to F throughout

Reviewer-2

1. For preemptable VMs, can someone ask for a cluster of VMs? It will be interesting to find if the cluster size impacts the probability of the VM preemption.
Revision made: New Figure 3b added to evaluate this scenario. Cluster-size does impact preemption rates, but there is no clear correlation.

2. Observation 3 is worded in too broad terms. I felt that “universal characteristics of preemptable VMs” is too strong that I did not see enough evidence to back this claim up. You looked at one provider and for a short amount of time. I suggest adding qualifiers to this claim.
Revision Made: Weakened observation 3.

3. Please clarify your assumptions for your lifetime and makespan analysis. Many applications (e.g., web services and batch data processing) in which failing only effects the currently running task in which case, the framework is resilient to single node failures (either due to reexecuting the lineage of the task or through repeating the task). For those cloud workloads, failures are not as painful. So your scheduling and checkpointing optimizations are mainly usefull for long running science applications in the cloud. It will be good to clarify this in the paper.
Revision Made: Workload assumptions paragraph added at the start of Section 4.

4. For figures 10, 11, it is not clear how many times did you run the experiment. Some of the results in figure 11 are too close to easily say a system is better than the other without a proper statistical analysis.
Revision Made: We added text in the eval section---we run each experiment atleast 5 times and report the averages.

5. The evaluation with ExoSpere came as a surprise, since the paper did not present this system as an alternative earlier. ExoSphere’s approach seems to work really well, so please discuss it earlier and clearly differentiate your approach than its.
Revision Made: We now explain exosphere’s relevance and how it compares in the eval section.

6. In Figure 10, Shapes does not seem to benefit from additional CPUs, can you please clarify that.
Revision Made: Added the following text in 6.2:
“This experiment does not measure scaling behavior, but instead highlights the differences between choices of VMs. On the other hand, the Shapes application can scale to a larger number of VMs, because the application does not have any significant communication overheads, and thus does not see any significant change in its running time when deployed on different kind of VMs.”

7. Also in figure 10, it seems (not sure) that there is also a correlation between the task execution time and the recomputation time. Check the results with code 4 and 8. I think it is worth mentioning this observation as well in addition to the correlation of VM size and the recomputation.
Revision Made: Added clarification in Section 7.2: “This recomputation time is roughly proportional to the original running time, in accordance with Equation 7”

8. The evaluation in Section 6.4 is interesting but I found it inconclusive without additional experimentation. First, it was not discussed on what resource SciSpot was deployed during this experiment. Also, did you run the entire trace from LANL? Did you run all the tasks from the trace? Also, what is in these traces?

Revision Made: In Section 6.4, we have added that “We emphasize that our goal is to compare the expected waiting time in HPC clusters vs. SciSpot's preemption overheads. The exact waiting time and performance is dependent on the specific HPC scheduler and hardware: we merely want to highlight that the performance overhead of preemptible VMs is reasonable if users do not wish to wait for HPC resources.”

9. You can not compare the waiting time or performance of two techniques on different hardware and operating conditions. The results will not be conclusive. Since the scheduler used in the traces is most likely open source, the authors should deploy the HPC scheduler and compare the two scheduling techniques on the same hardware.

Revision Made: Please see previous response.

10. When did you lunch the VMs? I mean, did you lunch VMs in batches every hour? Did you test with VMs with different size of memory.

Revision Made: Added at start of Eval section: “We ran more than 800 VMs running different bags of jobs in a ``closed-loop'' manner, such that SciSpot always launched a replacement VM to replenish preempted VMs. Since most bags of jobs take several hours to run, this gave us good temporal coverage of VM launches. All VMs of a bag of jobs were either run in the \texttt{us-east1-b} or the \texttt{us-central1-c} region.”

11. For the evaluation. The details of the experiment are missing. When did you run the VMs? How many VMs? In which region? How many preemptions did you face during your experiment?
Revision Made: Please see above.

12. I also suggest that you combine the background and related work sections. There is repetition between these two.
Revision Made: Merged related work into the background.

13. I encourage the authors to make their data set available to the public.
Revision Made: We have moved dataset URL from footnote to a new reference for easier discovery.

Reviewer-3

1. As the authors state in the submission, the main work was published at HPDC'20. It seems that there is no major work added work (i.e., 2.3 and 3.3 are added, but they do not make substantial contributions). Thus, I am not sure whether it is necessary to be published as an different article. I recommend that the authors add more works to make the current manuscript more suitable to be published as a separate publication, for example thorough theoretical analysis on bathtub distribution.
Revision Made: Thank you for comments. We have now added a paragraph (Novelty and Relevance) in the Introduction section, to explicitly note what new and expanded content has been added in this journal paper.

2. I found their observations are not founded from strong evidences (e.g., their empirical study of preemptions are made based on 870 VMs. However, it is not big enough to get a generalized observation).
Revision Made: 1. We have weakened some of our observations (#3 in Section 3, for instance). 2. We have included a subsection 3.4 on model validity and generalizability, and how even suboptimal models can be useful. 3. We have added additional data analysis of preemption data (Fig 3) to show that the temporal and concurrent analysis of bathtub preemptions. 4. Finally, the new discussion section tackles some more concerns and potential solutions to generalizability and long-term use of SciSpot.

3. Their assumptions are too big (e.g., they modeled the execution of scientific applications as "bag of jobs execution model." However, it is limited one only for computational job that explores a single combination of parameters. It is a huge assumption to make, and I do not agree that this can be a general characteristic of scientific applications. I think the range of contribution should be narrowed to "bag of jobs type of applications" and the title should be changed too.
Revision Made: We have added workload assumptions at the start of Section 4. Briefly, we require very coarse-grained job running time estimates for our job-scheduling/VM-reuse policy. These estimates are easier to obtain with bags of jobs, but other avenues of running time estimates can also be exploited (historical performance, analytical models, etc.). We have also expanded our explanation of transition-points policy in Section 4.2 to reflect these assumptions. We emphasise that bags of jobs typically explore a large parameter space and not necessarily a single combination of parameters.

4. The paper does not have conclusion or discussion section at all.
Revision Made: Added a Conclusion

5. The layout of footnotes may have better look.
Removed footnotes and included in text.